CHAPTER 3

CONTOUR CODING

This chapter develops contour extraction, selection and coding methods used in contour-based image coding system that we have developed in this study. First, how edge information is extracted from an image via generalized edge detector is explained. Then, we proceed with the edge following algorithm which produces long and smooth segments as much as possible. In order to increase the compression ratio while preserving the quality of the image in terms of visual appearance to human viewers and allowing small amount of degradation, some of edge segments which have less significance on surface reconstruction process are eliminated for a given threshold. Edge segments are ordered according to normalized weighted sum of their length, mean contrast and mean curvature. We have coded these contours by using *differential chain code* followed by Huffman coding. End points are coded in the form of distance between lexographicly ordered end points.

3.1 Introduction

Contours correspond to object boundaries where sharp changes occur due to some physical aspect of an image such as surface reflectance or illumination. It is well known that an image which consists of only its edge is highly intelligible and structural. Studies on human brain and low-level retinal processes have showed that brain also receives information in the form of edges and brightness. In fact, in

computer vision, intermediate and high-level processes such as shape from shading, structure from motion are partly based on edge information and so extracting the edges highly accurately has primal importance for any algorithm in machine vision. On the other hand, for compression purpose, contours have another important property that they are very sparse.

An edge detector extracting and locating object boundaries in an image from intensity data is a crucial step of contour-based coding system. The goal of edge detection is to obtain powerful and complete description from an image by characterizing these intensity changes. By using this representation, it would be possible to reconstruct a good replica of the original. Since accuracy of model parameters (contrast, intensity and width)[1] are determined by the accuracy of detected edges, edge detection is the most important part of the algorithm. An edge detector is expected to have a good detection, good localization, one response to one edge, robustness, efficiency, applicability to sparse data properties.

We have used generalized edge detector which allows the description of images on a plane called $\lambda\tau$–space where τ controls the shape of the filter and λ controls the scale of the filter. One can obtain most of the well-known edge detectors by setting these two parameters appropriately.

3.2 Generalized Edge Detector (GED)

Generalized edge detector [2] is based on the regularization theory and convolution with filters. In regularization theory, the smoothness constraint is imposed to the solution in by means of energy functional containing derivatives of the solution. GED uses hybrid energy functional :

$$E(f) = \int\int (f(x,y) - d(x,y))^2 + \lambda\,[(1-\tau)(f_x^{\,2} + f_y^{\,2}) + \tau(f_{xx}^{\,2} + 2f_{xy}^{\,2} + f_{yy}^{\,2})]\,dxdy \quad (3\text{-}1)$$

[1] Definition of these parameters and how they are extracted from an image will be explained in Chapter 4

In the functional, first term on the right hand side is a measure of the closeness of the solution *f(x,y)* to the data *d(x,y)*, and the second and the third terms are stabilizers on the solution as the first and second order derivatives. The Euler-Lagrange equation associated with this hybrid functional is

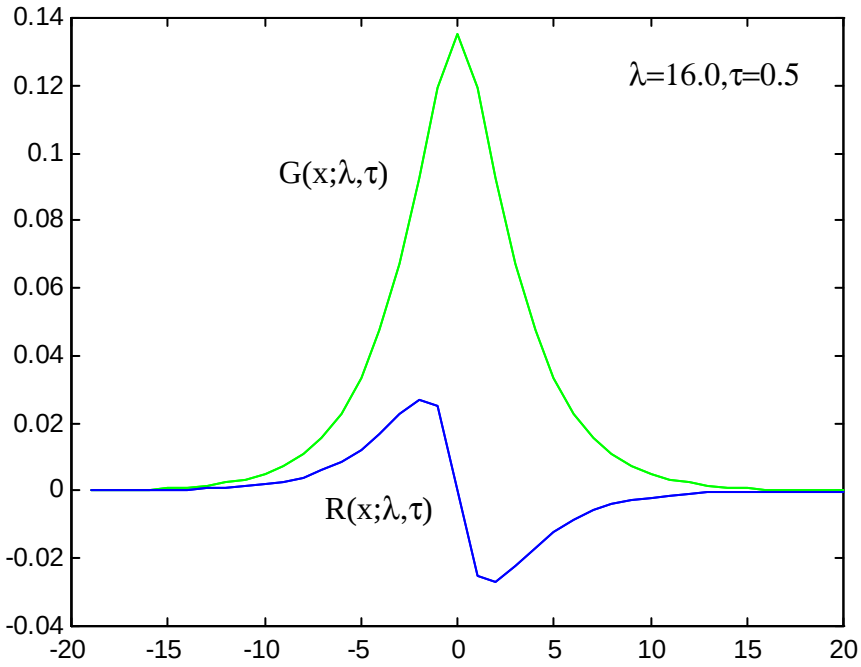
$$\lambda\tau\ [f_{xxxx} + 2f_{xxyy} + f_{yyyy}] - \lambda(1-\tau)\ [f_{xx} + f_{yy}] + f = d \qquad (3\text{-}2)$$

This is a forth-order partial differential equation. Solution of (1-2) is explained in details by using Green function in [2]. The solution gives five different form of filters denoted by G(x;λ,τ) and R(x;λ,τ) according to $\Delta = B^2 - 4A$, B and A where $B = \lambda\ (1-\tau)$ and $A = \lambda\ \tau$. These filters, G(x,λ,τ) and R(x;λ,τ), are drawn in Figure-3.1. Images are convolved with the filter G(x,λ,τ) and then edges are obtained from convolved signals by applying histerezis thresholding along tangent direction followed by thinning process. Images and edges detected for different λτ values are shown in Figure-3.2. G(x,λ,τ) and R(x;λ,τ) functions for five different cases are given in Table 3-1.

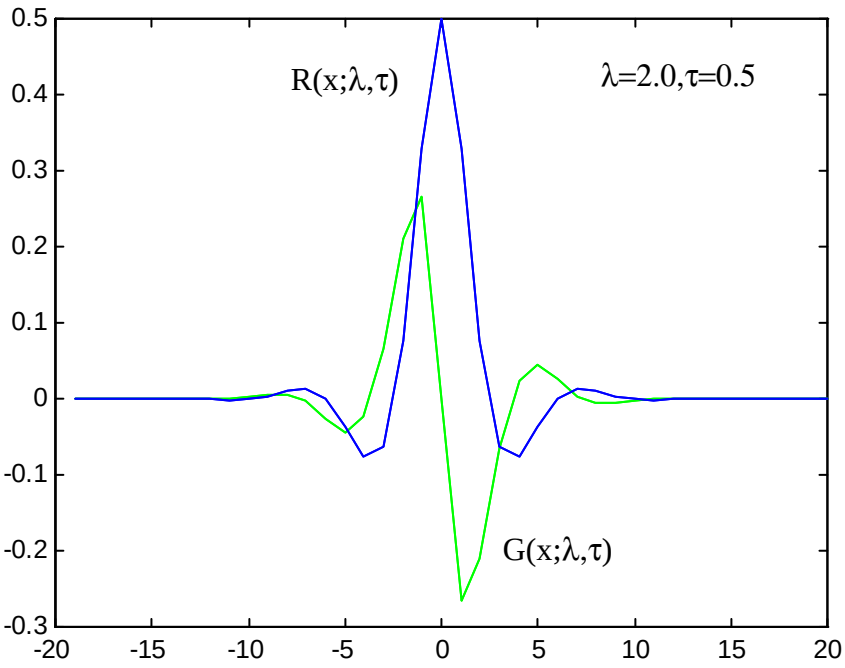
Case	G(x;λ,τ)	R(x;λ,τ)

Table 3.1 G(x,λ,τ) and R(x;λ,τ) filters

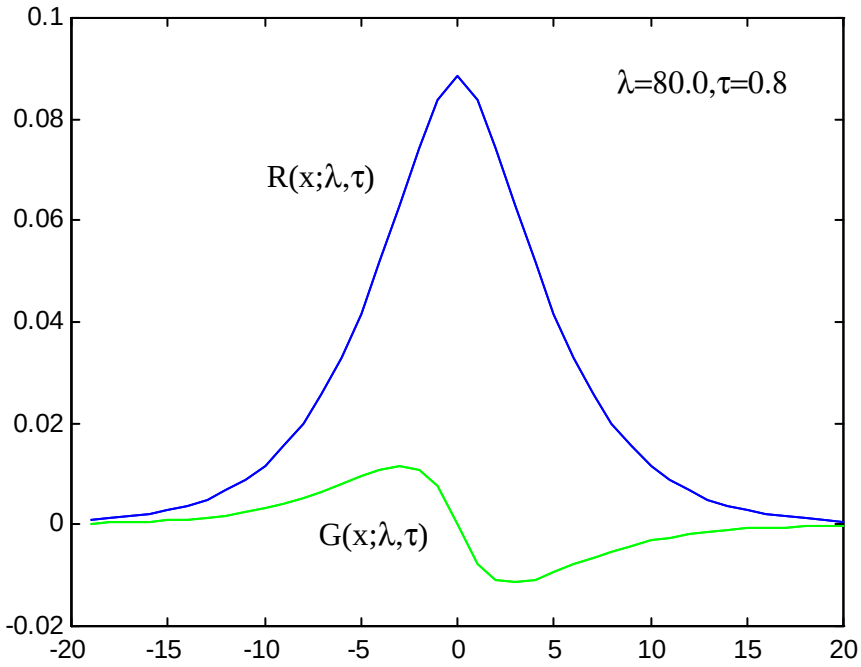
		$2A(b-a)\left[\frac{a}{\quad}\quad\frac{b}{\quad}\right]$
where $a=\sqrt{\frac{B+\sqrt{B^2-4A}}{2A}}$ $b=\sqrt{\frac{B-\sqrt{B^2-4A}}{2A}}$		
2. Δ <0	$G(x;\lambda,\tau)=\frac{-1}{4Aab}e^{-b\lvert x\rvert}\sin(ax)$	$R(x;\lambda,\tau)=\frac{e^{b\lvert x\rvert}}{4\sqrt{A}}\left[\frac{\cos(a\lvert x\rvert)}{b}+\frac{\sin(a\lvert x\rvert)}{a}\right]$
where $a=\frac{1}{A^{\frac{1}{4}}}\cos(\frac{1}{2}\arctan\sqrt{\frac{4A}{B^2}-1})$ $b=\frac{1}{A^{\frac{1}{4}}}\sin(\frac{1}{2}\arctan\sqrt{\frac{4A}{B^2}-1})$		
3. Δ =0	$G(x;\lambda,\tau)=\frac{-a}{2B}xe^{-a\lvert x\rvert}$	$R(x;\lambda,\tau)=\frac{e^{-a\lvert x\rvert}}{2B}\left[\frac{1}{a}+\lvert x\rvert\right]$
where $a=\sqrt{\frac{B}{2A}}$		
4. B=0	$G(x;\lambda,\tau)=\frac{-1}{2\sqrt{A}}e^{-a\lvert x\rvert}\sin a\lvert x\rvert$	$R(x;\lambda,\tau)=\frac{e^{-a\lvert x\rvert}}{4\sqrt{A}a}(\cos a\lvert x\rvert+\sin a\lvert x\rvert)$
where $a=\frac{-1}{\sqrt{2}A^{\frac{1}{4}}}$		
5. A=0	$G(x;\lambda,\tau)=\frac{-\operatorname{sgn}(x)}{B}e^{\frac{-\lvert x\rvert}{\sqrt{B}}}$	$R(x;\lambda,\tau)=\frac{e^{\frac{-\lvert x\rvert}{\sqrt{B}}}}{\sqrt{B}}$



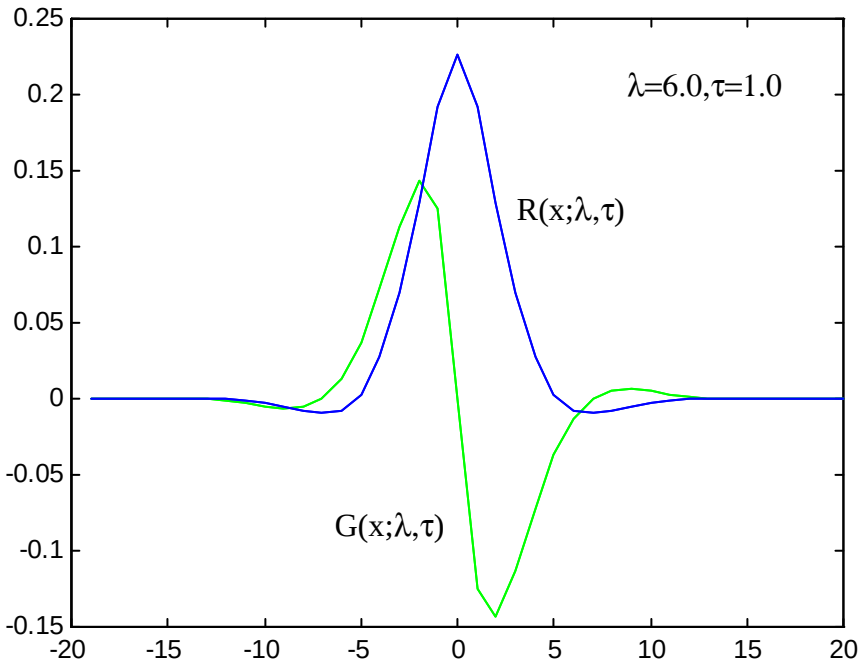
(a) Case 1, $\Delta > 0$



(b) Case 2, $\Delta < 0$



(c) Case 3, $\Delta = 0$



(c) Case 4, B=0